

Physically Based Cloth Shaders

Physically Based Cloth Shaders is a collection of powerful AAA cloth shaders based on Ashikhmin BRDF.

The shader supports translucency, retroreflectivity and anisotropic highlights.

The packages consists of 2 shaders:

- **Standard Cloth Shader:** The input is much like Unity Standard Shader but with another map slot for Retroreflectance and translucency. The shader provides 2 different BRDF models. The standard one and the other, Ashikhmin BRDF.
- **Masked Cloth Shader:** This shader can be used to blend between different materials, like cotton, velvet, satin and even metals. It blends between the above two mentions BRDF models based on the Retroreflectance mask.

Features:

- Retroreflective BRDF
- Can be used to simulate many kind of cloth materials as well as metals and plastics using masks.
- Anisotropic highlights.
- Support tiling Detail textures and normal maps.
- Translucency scattering support.
- Roughness Extrapolation Control(greatly increases sharpness in the roughness)
- Control reflection amount and control reflection with the ambient light through probes or lightmaps, thus preventing high reflection in darker areas.
- Opaque, Transparent and Alpha Cut support.

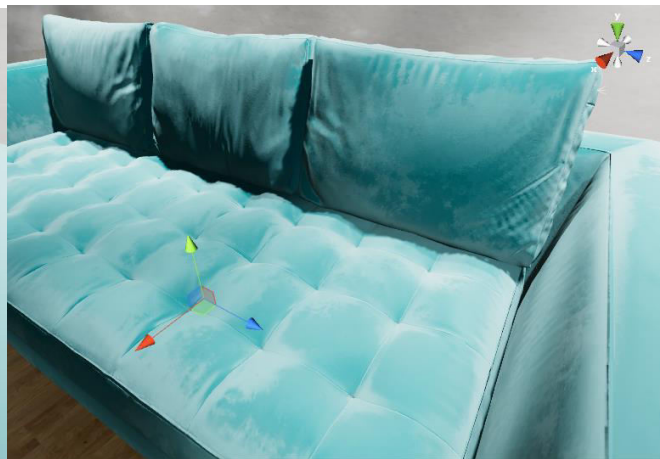
How to Use

Physically Based Cloth Shaders is very similar to Standard Shader except:

Retroreflectivity



Retroreflectivity: 0



Retroreflectivity based on mask

Both cloth shaders differ from each other in how they blend different BRDF models.

The shader **Standard Cloth Shader** doesn't blend between the Cook Torrance lighting model and Ashikhmin BRDF. The shader **Masked Cloth Shader** blends between the two BRDF models based on retroreflectivity mask. So in some sense the Standard Cloth Shader is cheaper on performance as compared the Masked cloth shader but the Masked Cloth Shader can be used to provide different variations in the same cloth material.

The Masked Cloth Shader expects you to provide a retroreflectivity mask where different areas of cloth contain different information. If you have different material on the cloth like metal, plastic and cotton, then paint the retroreflectivity mask as black in metal and plastic areas and paint the cotton area with some value or noise to achieve different material look.

Like you have metal carvings on the cloth then Masked Cloth Shader will be used where the metal is masked black in retroreflectivity mask.



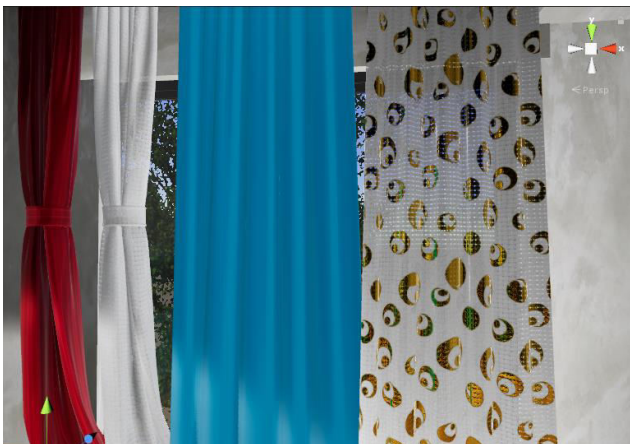
Retroreflectivity mask



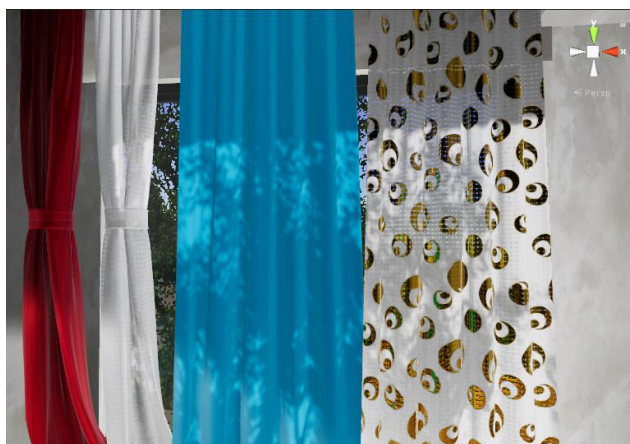
Final blend between velvet and metal surfaces where the metal used the Cook Torrance lighting model and the velvet used Ashikhmin BRDF

Translucency:

Translucency goes in the Alpha Channel of Retroreflectivity mask. Should be used for transparent cloth materials where you want the light to pass through the surface like transparent curtains and window blinds.



No translucency



Light passing through the surface

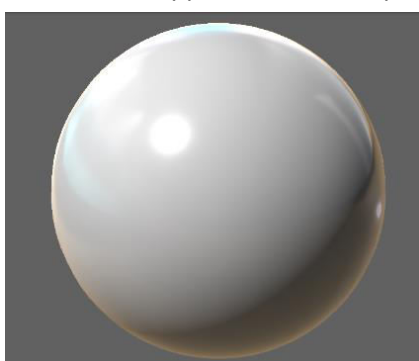
Also the shader supports alpha blending and opaque surfaces, to allow receive shadows on transparent surfaces set the rendering mode to transparent and Queue to AlphaTest+1 to prevent proper blending and shadow receiving:



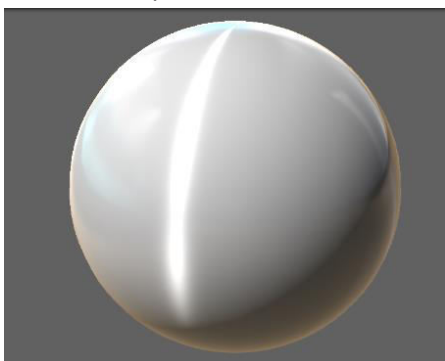
Transparent shader with blending with AlphaTested surfaces and shadow receiving and translucency

Specular Model:

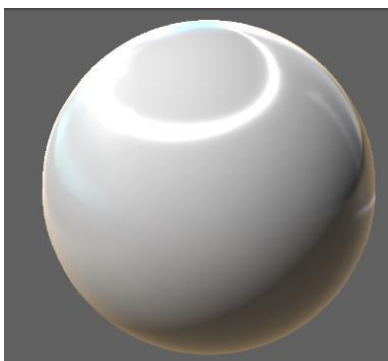
The shader supports GGX isotropic and anisotropic surfaces



GGX isotropic



GGX Anisotropic with rough on Y direction



GGX Anisotropic with rough on X direction

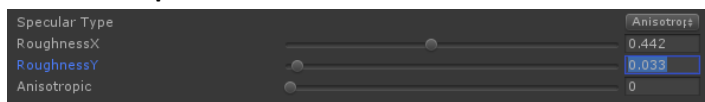
For isotropic surfaces:

The roughness of the material when using Standard(Isotropic) Specular need to be set at:



You can go into <1 in the Low values and >1 in the High value to get large amount of roughness contrast.

For anisotropic surfaces:



You have to set different values in roughnessX and Y parameters to get the desired result.

Also when using anisotropic surfaces the shader doesn't read the roughness texture. It is solely based on values. But the environment reflection(cubemap) can be blurred or sharpened based on the roughness texture and Roughness Extrapolation Control specified above.